AFRL-RI-RS-TR-2009-100
Final Technical Report
April 2009

DYNAMIC GAMING PLATFORM (DGP)

Lockheed Martin Corporation

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2009-100 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

JAMES M. NAGY
Work Unit Manager

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
APR 09	Final	Jul 07 – Mar 09

4. TITLE AND SUBTITLE

DYNAMIC GAMING PLATFORM (DGP)

5a. CONTRACT NUMBER
FA8750-07-C-0186

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Sergey Malinchik and Janet E. Wedgewood

5d. PROJECT NUMBER
PAIN

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
05

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Martin
Advanced Technology Laboratories
3 Executive Campus
Cherry Hill, NJ 08002-4103

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIED
525 Brooks Rd.
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2009-100

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2009-1382

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of the ProActive INTellience (PAINT) team's efforts was to construct two models of an example target system, consisting of leaders who controlled pathways that required resources and time to execute. Lockheed Martin designed strategies that represented probes that were ultimately implemented as changes to the model inputs. The degree to which the outputs of the two models differs is considered the diagnosticity of the probe(s). A Strategy Engine was also developed that used a Genetic Algorithm (GA) to develop probes and a Controller to interface the GA to the two PAINT models. We were able to increase three-fold the diagnosticity of the probes.

15. SUBJECT TERMS
Strategy, search,genetic algorithm,probe discovery,Proactive Intelligence,target system,strategy engine,search algorithm

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			James M. Nagy
U	U	U	UU	24	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

This work benefits from the efforts, insights, and inquiries of a number of people. The Government Team includes Dr. Peter Brooks, from IARPA; and our COTR, Mr. Jim Nagy. The Contractor Team included Mr. Mark Hoffman (PM), Dr. Sergey Malinchik (co-PI), Janet E. Wedgwood (co-PI), Mr. Tim Seidlecki (Technical/Integration Lead), Ms. Jennifer Lautenschlager (Systems Engineer), Dr. Steven Bankes (BAE), and David Groves (Evolving Logic). Dr. Ed Waltz and his colleagues at BAE were incredibly supportive in helping us integrate our efforts into the larger PAINT experimentation. We also wish to thank Dr. Ted Senator and Lars Hanson from SAIC.

1. EXECUTIVE SUMMARY

The ProActive INTellience (PAINT) program as a whole is exploring the hypothesis that simulation models of target dynamics of interest can be algorithmically explored to allow more rapid and accurate discovery of diagnostic probes that will reveal the actual functions of target systems and the intentions of their leaders, compared to current manual methods. The Lockheed Martin Advanced Technology Laboratories (LM ATL) effort provides the Probe Strategy Development, as shown at the top of the program-developed diagram in Figure 1[1].

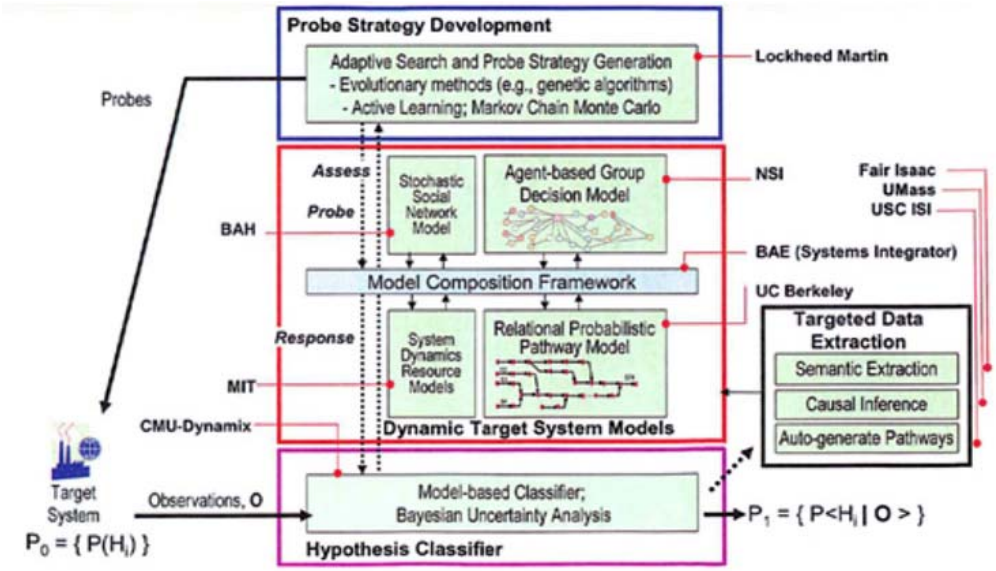

Figure 1: Overall PAINT functional architecture showing the LM Probe Strategy Development

The objective of the PAINT team's past year's efforts was to construct two models of an example target system, consisting of a pathways model (interconnected process segments that require resources and time to execute), leadership and social network models controlling a resource flow model and execution of segments in the pathways model. Furthermore, the PAINT team designed strategies that were represented as probes consisting of one or more actions that were ultimately implemented as changes to the model inputs. The degree to which the outputs of the two models differ in response to a probe is considered the diagnosticity of the probe. The goal of the LM ATL Strategy Engine is to discover highly diagnostic probes through efficient search in the input space of highly complex models.

[1] Demonstration 3 (D3) Plan, BASELINE, Ed Waltz, BAE Systems, September 8, 2008.

The challenge was to find an efficient search method that could discover robust strategies that discriminate among important model outcomes in a domain with many complex parameters. In particular, it was found that the models' landscapes are represented by rough multidimensional surfaces with many scattered maxima. The most appropriate search method was determined to be the Genetic Algorithm (GA).

The LM ATL team designed and developed the Strategy Engine, consisting of a Controller and an Optimization Engine as shown in Figure 2. The Optimization Engine generates probes and injects them into the models through the Controller. The Controller runs the hypothesis models, retrieves the measure of diagnosticity (fitness value) from the Hypothesis Comparator and passes a fitness value to the Optimization Engine. The Optimization Engine evolves the probes using built-in operators and heuristic rules developed by LM ATL. This process continues until the number of iterations (generations) reaches the predetermined threshold.

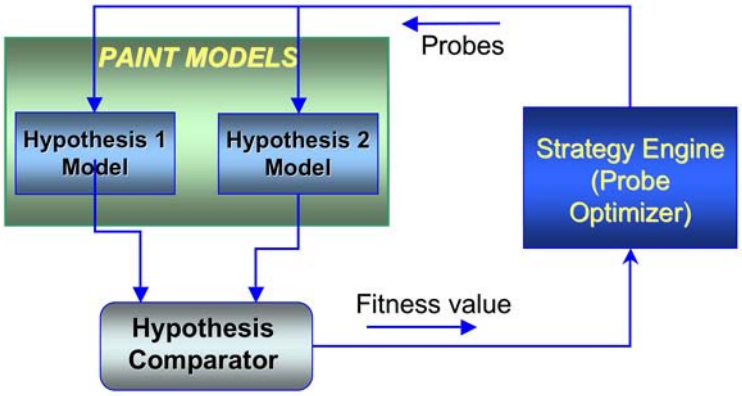

Figure 2: The Strategy engine uses a GA to search the probe space

There are many metrics, but the top three were the:

- Growth of the fitness function which is correlated with improving diagnosticity of the probes.
- Number of iterations needed to discover and converge to most significant maximum.
- Balance between Exploration versus Exploitation: keep extensive search strategy while improving the discovered probes.

The LM ATL team has shown the effectiveness of using the GA search method, combined with a flexible controller to perform optimization functions in the PAINT Strategy Engine. We were able to develop a domain specific GA with unique collection of operators and selection functions which demonstrated very high efficiency in evolving the probes. The steady increase in the diagnosticity measures as the probes were tested on the models is shown in Figure 3. The number of possible strategies for this particular set up is shown in the circle on the left. The displayed strategy consisted of four actions that affected ~15 parameters that were set to particular values by the GA. This is a result that would have been extraordinarily difficult and time consuming, if not impossible, to find using other, more simple methods (Monte Carlo). Typically, after about 1200 iterations, diagnosticity of the probes was improved by factor of three (from 14,000 to 42,000). The analysis of the best probe evolution tracks indicates that a good balance between Exploration vs. Exploitation (keeping an extensive search strategy while improving the discovered probes) has been achieved.

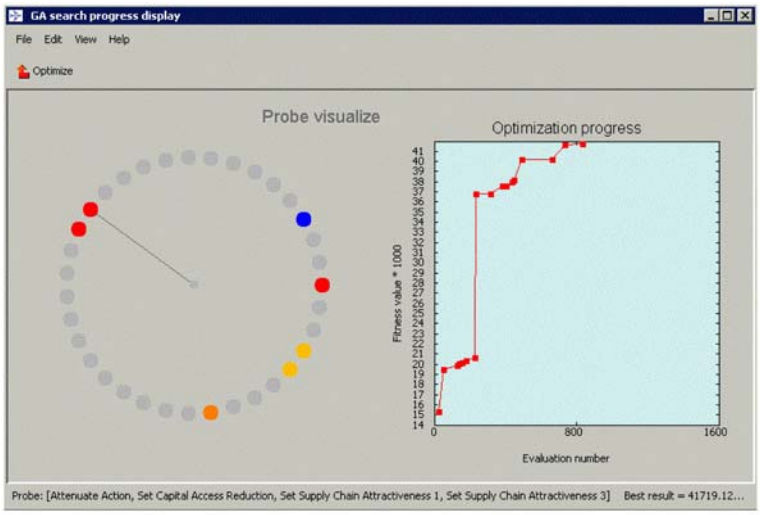

Figure 3: The GA produced results that would have been difficult to find using other methods

A few of the other areas of research, more of which are detailed in the report, include:

- Understanding how to measure the third metric: Exploration vs. Exploitation.
- Exploring the nature of the probes that were discovered, including their relation to actual actions that could have been taken in the real world and observable indicators.
- Automation of some of the work needed to maintain the GA once it has been instantiated in a domain of interest.
- Probe Identification—experimental runs identify actionable parameters that can be understood as probe inputs.
- Increased efficiency in searching for probes and sophistication of probes—improved seeding of algorithms.
- Support for conditional probes, or strategies that consist of multiple probes.
- Development of visualizations that is more user-oriented.

2. INTRODUCTION

The PAINT program as a whole is exploring the hypothesis that simulation models of target dynamics of interest can be algorithmically explored to allow more rapid and accurate discovery of diagnostic probes that will reveal the actual functions of target systems and the intentions of their leaders, compared to current manual methods. The LM ATL effort provides the Probe Strategy Development, as shown at the top of the program-developed diagram in Figure 1.

The objective of the PAINT team's past year's efforts was to construct two models of an example target system, consisting of leaders who control pathways (interconnected process segments that require resources and time to execute). These models correspond to two different conceptual hypotheses as shown in Figure 4.

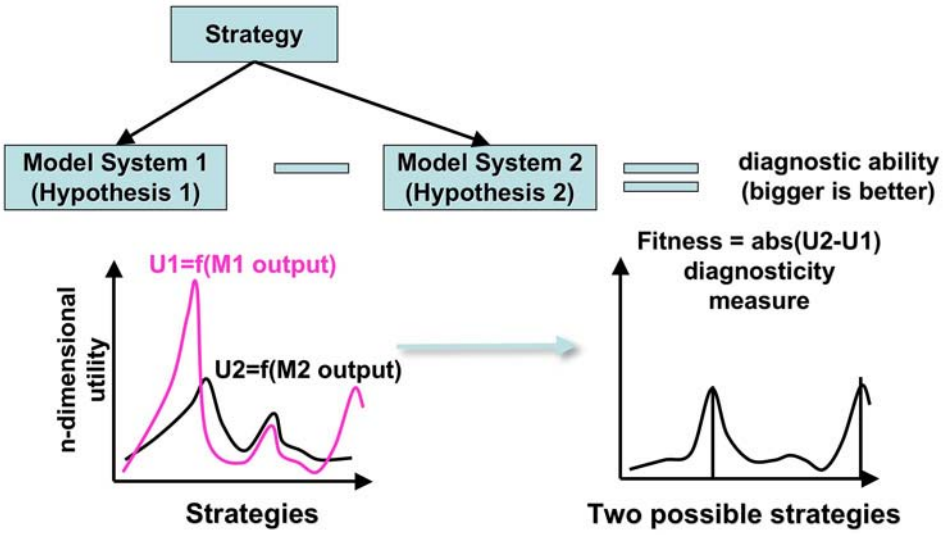

Figure 4: The LM ATL Probe Strategy Development seeks to develop probe strategies that differentiate between model hypotheses

The team of LM ATL and BAE is specifically addressing the search for diagnostic responses using our Probe Strategy Development effort for PAINT (Figure 1). In other words, we are searching for sets of model inputs to the system that will cause the two models to produce different outputs, thus differentiating them. The degree to which they are differentiated by the inputs identifies the diagnosticity of the inputs. To support discussion and development of these searches, the team has made the following definitions[1]:

- "*Probing Strategy*: High-level approach to conduct a conditional test of a target by stimulation (e.g., "Weaken ability to perform, cause correction")." This will be referred to simply as a "strategy."
- "*Probe*: A specific operational activity to stimulate the target—as an element of a strategy (e.g., "Restrict access to capital and materials by sanctions")." There may be more than one probe necessary to implement a probing strategy.
- "*Probe Implementing Action*: Detailed model inputs to implement a probe (e.g., "reduce available scientists by 50%; reduce operating capital and material by 25%")." This map directly to changes in the model parameters.

Our approach is to generate strategies that provide significant diagnosticity (are able to differentiate the models) through:

- Strategy Decomposition and Algorithm Selection:
  - Researchers run studies to understand the solution space and identify candidate strategy decompositions and search algorithms.
  - Researchers implement selected search algorithm(s) and provide controls and visualizations to support user input/guidance.
- Strategy Assessment:
  - User runs strategy search algorithm, examines results and provides focus for next set of searches.
  - System will include (although this is not yet implemented) sampling over input uncertainty to ensure that selected strategies are not near "danger" zones where small changes in inputs cause large changes in the outputs.

Under Strategy Decomposition, we determined that the GA was most appropriate to conduct the search in this complex, multi-dimensional space with many local maxima. Under Strategy Assessment, we ran the algorithm on the two models and discovered probes that were highly diagnostic. The uncertainty issues were handled by Carnegie Mellon University (CMU), therefore we did not do sampling over the input uncertainty, but rather used the CMU output that considered the uncertainty and included it in the fitness function.

The DGP leverages the LM Strategy Engine Core, consisting of the core GA Engine as the optimizer, and a Controller that interfaces to the PAINT Models through the Strategy API on the BAE backplane, as shown in Figure 5.

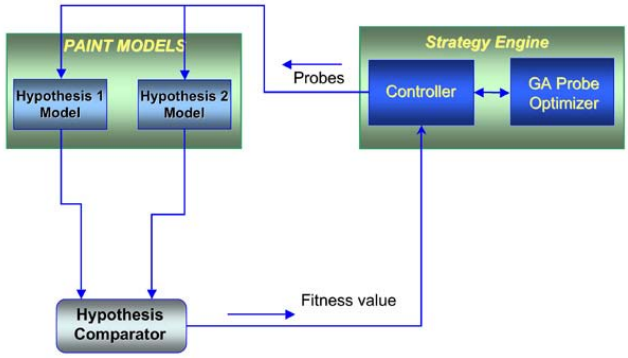

Figure 5: the Flexible Controller interfaces to the Models through the Strategy API, transparently handling changes to the models and actions for the GA Optimizer

The Strategy Engine, combined with the PAINT models instantiated on the BAE backplane, produced a capability that has never been available to the intelligence community. Specifically, they enable the analyst to:

- Intelligently explore wide ranging actions that could be taken on indeterminate situations.
- Discover counterintuitive solutions through combinations of probes and probe parameters.
- Illuminate characteristics of the situation that help separate the hypotheses in a timely fashion.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Selection and Implementation of the Genetic Algorithm (GA)

It is clear that in the case of finding probes/strategies using the Strategy Engine we have a typical global optimization problem with a multidimensional function in a very large search space. The function is generated by the response of a complex model to a particular set of parameter/input values. This space is by no means a nice smooth space. Therefore, we can immediately eliminate the classical methods like gradient descent, simplex, and quasi-Newton methods. These do not do well on rough surfaces with many local maxima.

Through running and observing the model outputs, we have been able to confirm our original assumption that the search landscape is represented by a rough multidimensional surface with scattered multiple maxima. There are many popular techniques for global optimization search for that kind of function. All of the methods are stochastic in nature and designed to avoid being trapped by local maxima. Some of these are: Simulated annealing, Stochastic tunneling, Particle swarm optimization, Ant colony optimization, Bees algorithm, and Evolutionary Computing (including GA and Evolutionary strategy).

The last class of methods, from our point of view, represents most of the powerful, flexible and efficient optimization techniques. The GA is not a single technique, but rather a large family of techniques. Its flexibility lies in the large number of methods that can be used to generate the "next" generation of probes. In other words, given that we are able to develop the proper smart/sophisticated operators, heuristic rules, flexible constraints, and human interaction with the ongoing search process, the navigation through the complex search space will be very efficient and productive. The GA will work, on average better than these other techniques. The Co-PI on this program, Dr. Sergey Malinchik, co-authored a book on this subject (Malinchik, 2004)[2].

A criticism leveraged against the GA is that creating domain specific chromosome structures, operators, selection rules and other attributes is more of an art than a science, and that done poorly, can lead to unsatisfactory results. We believe that we are responsive to this criticism through the development of a GA toolkit that provides valuable (visual) insight into the progress of the GA as it is evolving new solutions that assists in the design. In future versions, it will also enable the user to have reasonable control over the mutation range of genes, fitness function, and constraints weights. The ability of the user to view the results of a current generation and hand select candidates to guide the search, provides a powerful human aspect to the process.

[2] Bandte, Oliver, Malinchik, Sergey, " A Broad and Narrow Approach to Interactive Evolutionary Design – An Aicraft Design Example", Springer Berlin/Heidelberg, June, 2004.

The key to the success of the GA is the careful selection of the operators and the search space representation. For the DGP, we designed a chromosome, which is essentially a probe, to contain between one and five different actions. Each action is characterized by a set of genes that are the parameters. Each action contains its own specific parameters, as well as a start time and duration. Figure 6 provides an example of a chromosome and several genes.

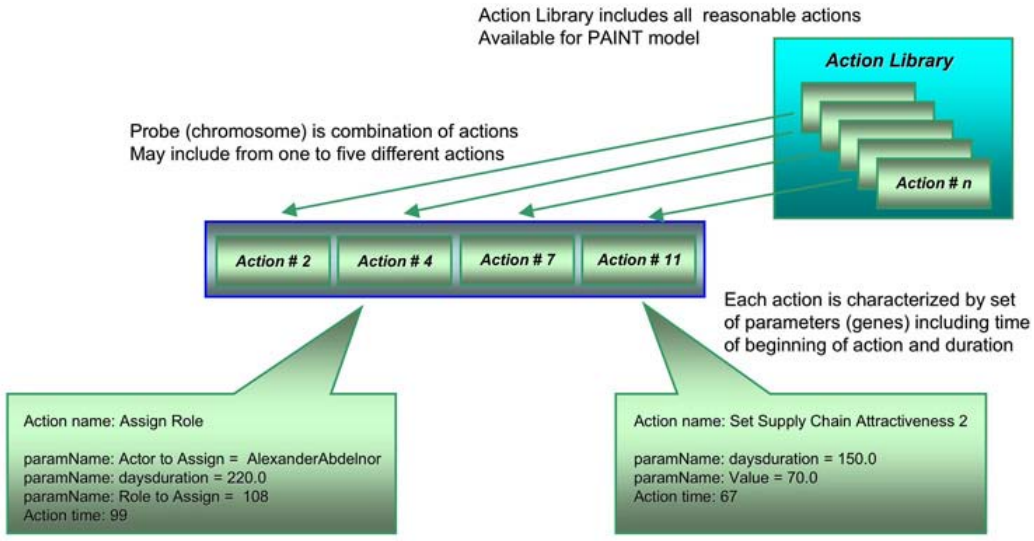

Figure 6: Genes for the "Assign Role" and the "Set Supply Chain Attractiveness 2" actions

The GA operators are used to manipulate the chromosomes and genes that are used to produce the next generation (offspring) of the GA evolution. Figure 7 shows three types of operators that were used. For each new generation, one of the operators is randomly selected with equal weighting. Further research would be required to determine if a different operator selection algorithm would produce better/faster results.

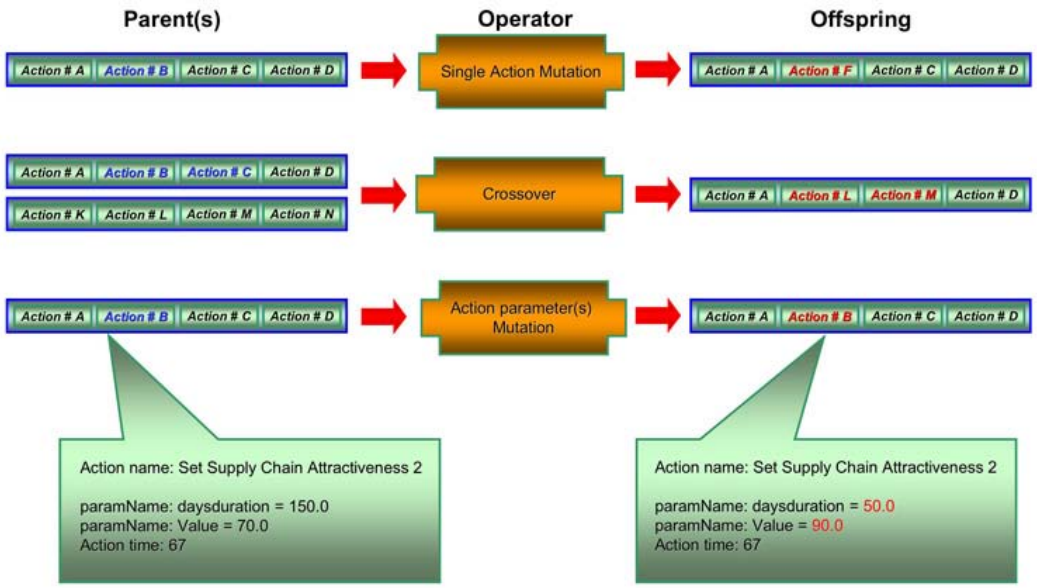

Figure 7: Three different GA operators that were used to produce the next generation of probes

3.2 Controller

The block diagram of the system is shown in Figure 2. The Controller runs in conjunction with the BAE Model Backplane and the two Hypothesis models. The primary focus for the Controller architecture was to provide flexibility to reduce the amount of recoding needed as the models, the actions and their parameters changed. This allowed us to run the GA on almost continuous releases of the code. We also implemented a module that randomly selected and modified actions that we used until the GA was available. This early capability helped find several issues with the modeling and action code/data. The controller allows a developer who is not familiar with the Model Backplane or how to run the models within (initializing, making a plan based on probes, etc.) to easily modify probes, run the model using these probes, and finally access the fitness value. A developer only needs to have basic knowledge of collections in Java, which the probes are stored in as a List, as opposed to the detailed and large hierarchical class structure of the Strategy Engine's API for modifying probes.

The Controller has simple API commands to initialize and run the PAINT models, insert a plan (set of probes) and acquire results. The plan parameters can be generated by the GA, or any other mechanism that generates the appropriate data. Therefore, should we decide to use a different search algorithm, we simply swap out the GA and swap in the new algorithm.

The operation of the system is as follows:

- Initializing the Controller:
  - Locates and acquires hypothesis models on the Backplane based on the model's name.
  - Loads and stores the model's possible actions with the action's parameters into a collection using our ModelLoader:
    - ModelAction contains the action (probe) data and the associated parameters (ModelParameter).
    - ModelParameter contains the parameter data.
    - Both ModelAction and ModelParameter contain randomize functions for time and value data, if the GA is not available.
- Modifying Actions and Parameters:
  - The Actions and Parameters are sent to the GA module where they are modified according to the selected operators. The GA modifies the parameters simply by using the "set" function based on the type of parameter (numeric or enumeration).
  - Actions not used are removed from the Action collection.
  - Actions used are sent back to the controller.
- Running the models through the Controller:
  - Based on the final actions chosen, the action collection is converted into a PAINT Strategy Service's Plan:
    - The Plan is then written to the backplane using the Strategy API. It is executed as each model runs.
  - Once the models are finished, the controller then executes the fitness evaluation function and stores the fitness result for the GA to access.

4. RESULTS AND DISCUSSION

We developed a variety of metrics that measure how efficiently the GA is performing its search, and how widely the GA search is covering the space. The metrics shown in Table 1 were developed in July 2008. An additional metric called "Exploration vs. Exploitation," i.e., how to keep extensive search strategy across the space while improving the discovered probes was added in January 2009.

Table 1: Indicates that the GA is a viable solution for searching the probe space

Draft Metrics for the LM Strategy Engine	Phase I Goal	Sample: Jan 2009
Number of strategies evaluated (N): This is a very raw number—you might evaluate 10M strategies and find none that are diagnostic. Also, over time, a lower number might mean a more efficient system. This metric can be affected by: • Improvement from seeding. • Time to evaluate probes – depends on the time it takes the models to execute. • Time required for users to intervene.	150 in a session	1200 with no user intervention
Number of useful strategies found/number evaluated: • How many different results does the system suggest that the user contemplates?	2%	N/A Although the probes provided increased fitness, we did not have the SME input to determine the usefulness/ actionability.
Number of additional factors considered in search: • Such as cost, risk, number of people involved, time to complete, etc. • Very dependent on the data being available.	0	0
Diagnosticity (D): How well are we differentiating between the systems: • Percent increase over time on "similar" systems- This has a "moving target" problem—we will need to characterize the systems and make adjustments for the change in the system in order to come up with a compelling measure for this. • Complexity of probes may improve diagnosticity. • Number of probes tested might also. • Ability to discover good diagnostic measures.	0%	The diagnostic measure is the fitness value from the CMU model. Early tests started out at about 18,000 for the fitness value. Final best value was around 41,000. Complexity of probes improved diagnosticity. We were limited to one probe, although we could have multiple actions per probe.
Robustness (R): The ratio of robust strategies to total strategies: • By evaluating more probes, you are more likely (but not guaranteed) to find more robust ones. How do we avoid "refinding" non-robust probes? • May be improved by some type of filtering based on known cliffy areas.	1%	This measure was incorporated into the CMU model fitness function, and is not available separately at this time.

Acceptance (A): Where does the user fall on the spectrum of completely allowing the tool to provide probes to not using the tool at all, we will develop the following factors into a metric • Number of times the tool is invoked. • Number of redirections by the user—need to understand this, whether more or less is better? • Number of times analyst uses the results. • Number of times the analyst lets the system suggest strategies (see next metric).	Low-New, not used to it, interface may not be friendly	This metric cannot be evaluated until there are users involved
Number of new useful strategies discovered/ strategies used (the – "oh, I didn't think of that" factor): • This is the reverse of the current method where the user tells the system what the parameters of the probes are. Here, the system finds combinations of inputs that are highly diagnostic, then the user or the system and the user discover strategies that cause those input combinations. • The value is the number of strategies the system develops for which the user can determine an operational equivalent.	0	This was not a phase I task.
Number of "identifiable" strategy clusters in the Strategy Library (nPSL): • Basically, the number of **Strategies** that are statistically different from each other in the library based on a multi-dimensional locality measure that we will define. • Highly dependent on system stabilization (or things won't cluster). • This will assist in your seeding capabilities, so a higher number of clusters means more granularity and possibly faster convergence.	0	This was not a phase I task.

We were not yet able to determine the operational values for all of these metrics at this time. Certainly, the speed with which we can develop a set of probes is a key to the success of any search algorithm. This value is dependent on the performance of the models (how long it takes to execute the models once) as well as how many iterations are needed for the GA to converge to a good answer (typically thousands of runs). The ability to seed the GA to a promising search area could reduce the number of runs needed. This activity, scheduled in a later phase, would have built up a library of probes and responses from which a smart algorithm could pick a reasonable starting point for the GA, as opposed to a random starting point.

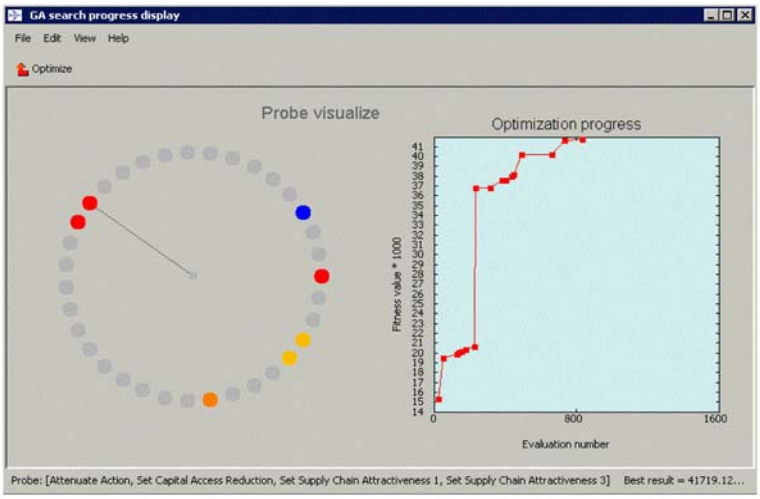

Figure 8: The right side shows the steadily increasing value of the fitness function as the GA evolves ever better probes

We have made significant progress, to the point that we were major contributors to the prototype demonstration. In Figure 8 we show the results of one of the highly diagnostic probes that were discovered by the GA. On the left side in Figure 8 is shown the set of possible 35 probes (small) circles arranged along the large circle), given that we were allowing four actions per probe from a total of seven available probes. Each probe (consisting of four actions where each action is characterized by three to four parameters) represents the subspace of the whole search space. During the search process the best current probe is marked by the radial line. The right side in Figure 8 shows the progress of the GA optimizer. The y-axis is the fitness (diagnosticity) of the solution while the x-axis is the iteration of the evaluation. The actual diagnosticity value of over 41,000 was reached in only about 1200 iterations starting from the value of about 15,000. Given more time, it is possible that a probe would have been found that produced even better diagnosticity. The actual real-world value needed for the diagnosticity to be operationally useful has not yet been determined.

As mentioned earlier, the GA will perform better than any of the other search algorithms when measured based on convergence rate, simply because it is an adaptive search method with an intelligent search algorithm that will always perform better than random.

5. CONCLUSIONS

Our Strategy Engine uses a Genetic Algorithm to develop probes (sets of model input values) and a Controller to interface the GA to the two PAINT models. The Controller design successfully abstracted the complexity of the Strategy API away from the GA, creating a modular environment that can be easily applied to additional problems.

We were able to develop a domain specific GA with unique collection of operators and selection functions which demonstrated very high efficiency in evolving the probes. Typically, after about 1200 iterations, diagnosticity of the probes was improved by factor of three (from 14,000 to 42,000). The analysis of the best probe evolution tracks indicates that a good balance between Exploration vs. Exploitation (keeping an extensive search strategy while improving the discovered probes) has been achieved.

Working with the GA in a modeling and simulation environment consisting of brand new models is a challenge. We have been involved in several similar programs and are well aware that the models are constantly being updated. These updates are usually not limited to the internal algorithms, but often changes/adds/deletes inputs, outputs and parameters. This makes it challenging for the GA developer because in order to build a good GA, one must have both knowledge of the model parameters, to define the representation of the search space (chromosomes and genes) as well as the generated fitness function, to understand the most effective use of the operators (mutation, etc.). By taking many versions of the models as they evolved, we were able to gradually learn about the models solution space and improve on the GA, even though the models were not yet completed.

In general, our work demonstrated that the GA-based probe search and optimization technique is a very effective and promising approach allowing us to improve quickly existing solutions and produce completely new counter-intuitive results

6. RECOMMENDATIONS

This program opened up many avenues of research as follows:

- How much of the expertise of designing the GA can be captured and made available as an assistant to the user: The use of the GA remains elusive to many due to the complexity of designing the solution space representation and the operators. It appears that once there is a reasonable representation, that an intelligent assistant could aid in tuning the GA. Furthermore, the assistant may even be able to run analysis on the models, and given information about which parameters are available to it, might even be able to assist in the design of the GA.
- Improved performance through intelligent seeding of GA: It may be possible to intelligently seed the GA by finding historical situations that are similar to the current situation in which an action was taken that produced desirable results. This would require the development of a library of linked situations, actions and results, as well as a method to measure the "closeness" of the historical situation to the current situation, and the "closeness" of the historical result to the current result.
- Improvement through weighting of operators: In the experiments that we did, the operators were weighted equally. However, additional experimentation would be expected to yield an unequal weighting that would provide faster convergence, while still satisfying the Exploration vs. Exploitation metric (see next).
- Exploration vs. Exploitation: This metric, which compares the amount of effort the GA is expending in searching different areas of the solution space vs. deeply exploring a particular bump in the space has rather broad implications. Too little searching around the space means there is a high probability that an important part of the space is not being explored. Too much searching means there is not enough local exploitation of interesting areas. Finally, there is no current expression of how to measure this ratio, and what qualifies as a good ratio.
- Improved performance through human interaction: Research has been done that indicates that putting a human in the loop to help focus the selection of the next generation of probes can greatly improve the convergence time of the GA. This capability has been used in previous work, but we were not able to include it in the time that we had.
- User-oriented visualization and relationship to the real world: The current visualization is oriented towards the understanding of the fitness function and its value. This value needs to be tied in a meaningful way to the user that indicates the state of the models that produced the fitness function. This would require some very close interaction with the user, to understand how they want to view such models, and also, what observables are available to the user in the real world and how they relate to the models and to the fitness function.

7. RELEVANT REFERENCES

Bonabeau; Eric, Anderson; Carl, Orme; Belinda, Funes; Pablo, Bandte; Oliver, Sullivan; Mark, Malinchik; Sergey, Rothermich; Joseph (2006) Methods and systems for interactive evolutionary computing (IEC), United States Patent 7043463, issued on May 9, 2006.

Malinchik S, Bonabeau E (2004) Exploratory Data Analysis with Interactive Evolution, Lecture Notes in Computer Science; 3103:1151-1161.

Malinchik S, Orme B, Rothermich J, and Bonabeau E (2004) *Interactive Exploratory Data Analysis*, Proc. of the 2004 IEEE Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ.

Bandte O, Malinchik S (2004) A Broad and Narrow Approach to Interactive Evolutionary Design - An Aircraft Design Example, Lecture Notes in Computer Science; 3103:883-895.

8. LIST OF ACRONYMS

CMU	Carnegie Mellon University
DGP	Dynamic Gaming Platform
GA	Genetic Algorithm
IARPA	Intelligence Advanced Research Projects Activity
LM ATL	Lockheed Martin Advanced Technology Laboratories
PAINT	ProActive INTelligence